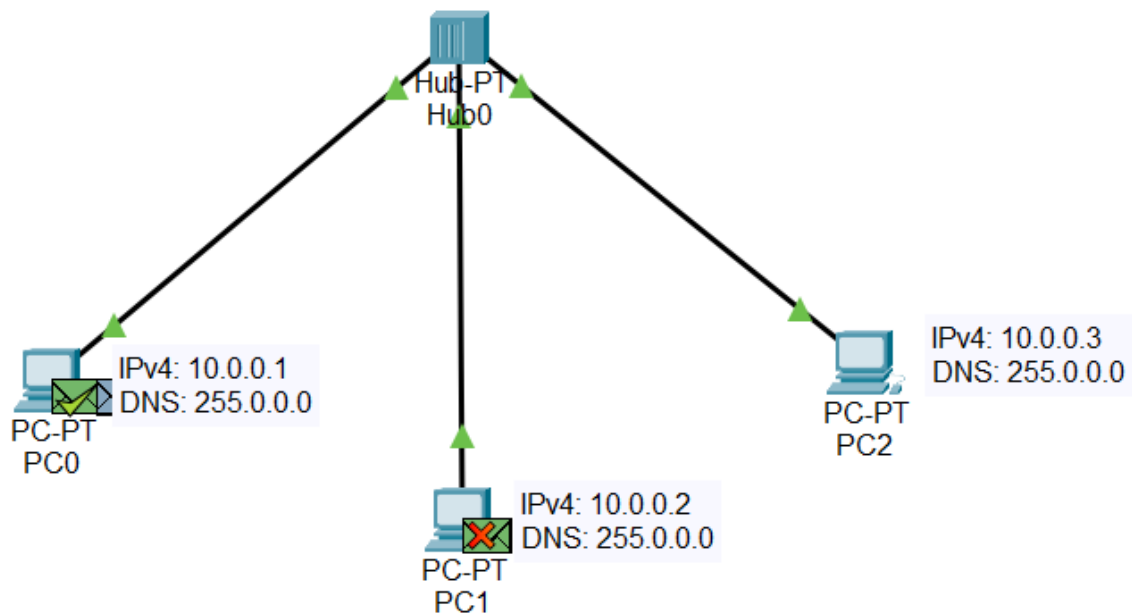




COMPUTER NETWORKS

LABORATORY PROGRAM – 1

Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC2	ICMP		0.000	N	0	(edit)	

```
C:\>ping 10.0.0.3
```

```
Pinging 10.0.0.3 with 32 bytes of data:
```

```
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
```

```
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
```

```
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
```

```
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
```

```
Ping statistics for 10.0.0.3:
```

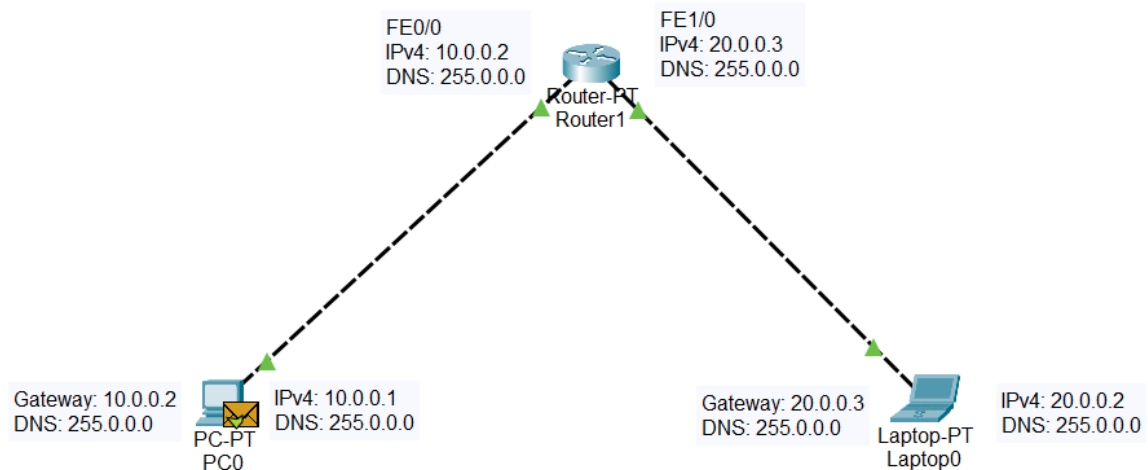
```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
    Approximate round trip times in milli-seconds:
```

```
        Minimum = 0ms, Maximum = 9ms, Average = 2ms
```

LABORATORY PROGRAM – 2

Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	
	In Progress	PC0	Laptop0	ICMP		0.000	N	1	(edit)	
	In Progress	PC0	Laptop0	ICMP		0.000	N	2	(edit)	

```
Cisco Packet Tracer PC Command Line 1.0
```

```
C:\>ping 20.0.0.3
```

```
Pinging 20.0.0.3 with 32 bytes of data:
```

```
Reply from 20.0.0.3: bytes=32 time<1ms TTL=255
```

```
Reply from 20.0.0.3: bytes=32 time<1ms TTL=255
```

```
Reply from 20.0.0.3: bytes=32 time<1ms TTL=255
```

```
Reply from 20.0.0.3: bytes=32 time<1ms TTL=255
```

```
Ping statistics for 20.0.0.3:
```

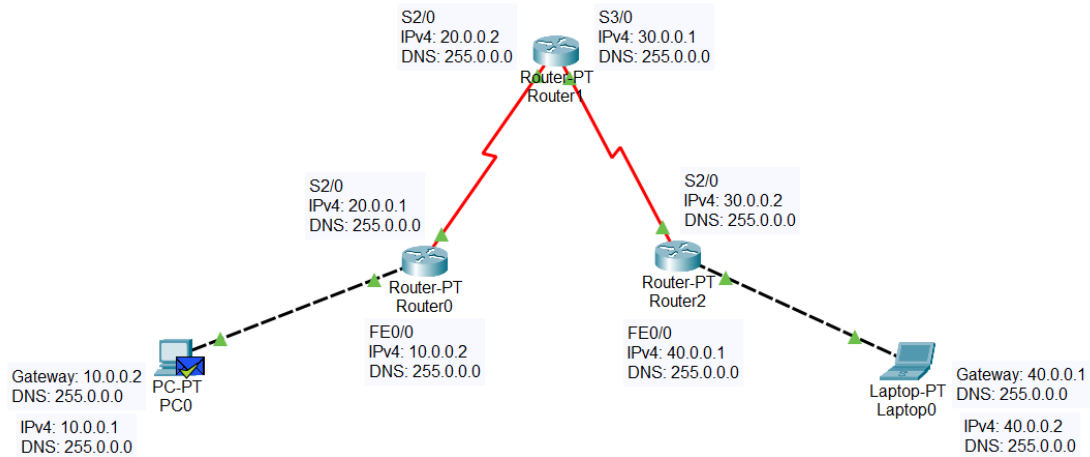
```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

LABORATORY PROGRAM – 3

Configure static route to the Router.



SHOW IP ROUTE

```
C 10.0.0.0/8 is directly connected, FastEthernet0/0
C 20.0.0.0/8 is directly connected, Serial2/0
S 30.0.0.0/8 [1/0] via 20.0.0.2
S 40.0.0.0/8 [1/0] via 20.0.0.2
```

Figure 3.1: Router0

```
S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.0/8 is directly connected, Serial3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2
```

Figure 3.2: Router1

```
S 10.0.0.0/8 [1/0] via 30.0.0.1
S 20.0.0.0/8 [1/0] via 30.0.0.1
C 30.0.0.0/8 is directly connected, Serial2/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
```

Figure 3.3: Router3.3

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 40.0.0.2

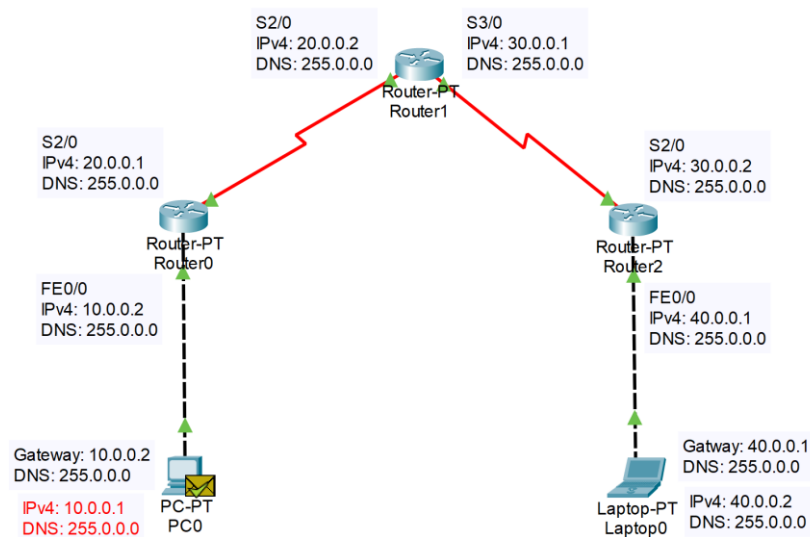
Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=36ms TTL=125
Reply from 40.0.0.2: bytes=32 time=34ms TTL=125
Reply from 40.0.0.2: bytes=32 time=30ms TTL=125
Reply from 40.0.0.2: bytes=32 time=26ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 26ms, Maximum = 36ms, Average = 31ms
```

LABORATORY PROGRAM – 4(A)

Configure default route, static route to the Router.



SHOW IP ROUTE

Gateway of last resort is 20.0.0.2 to network 0.0.0.0

```
C 10.0.0.0/8 is directly connected, FastEthernet0/0
C 20.0.0.0/8 is directly connected, Serial2/0
S* 0.0.0.0/0 [1/0] via 20.0.0.2
```

Figure 4.1: Router0

```
S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.0/8 is directly connected, Serial3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2
```

Figure4.2: Router1

Gateway of last resort is 30.0.0.1 to network 0.0.0.0

```
C 30.0.0.0/8 is directly connected, Serial2/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1
```

Figure 4.3: Router2

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	

```
C:\>ping 40.0.0.2
```

```
Pinging 40.0.0.2 with 32 bytes of data:
```

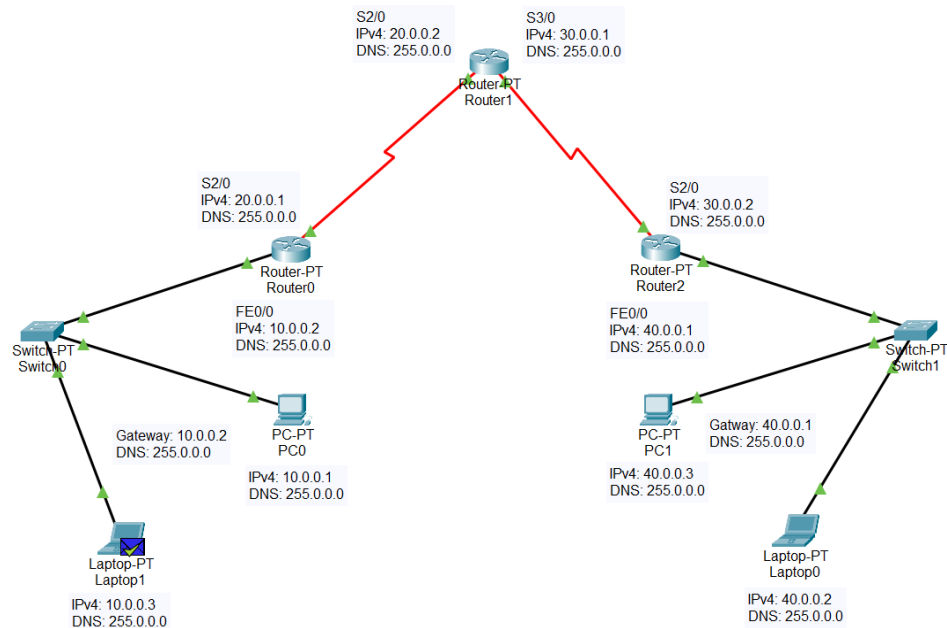
```
Reply from 40.0.0.2: bytes=32 time=34ms TTL=125
Reply from 40.0.0.2: bytes=32 time=33ms TTL=125
Reply from 40.0.0.2: bytes=32 time=30ms TTL=125
Reply from 40.0.0.2: bytes=32 time=33ms TTL=125
```

```
Ping statistics for 40.0.0.2:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 30ms, Maximum = 34ms, Average = 32ms
```

LABORATORY PROGRAM – 4(B)

Configure default route, static route to the Router, inclusive switches.



SHOW IP ROUTE

```
Gateway of last resort is 20.0.0.2 to network 0.0.0.0
C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial2/0
S*   0.0.0.0/0 [1/0] via 20.0.0.2
```

Figure 4.1: Router0

```
S    10.0.0.0/8 [1/0] via 20.0.0.1
C    20.0.0.0/8 is directly connected, Serial2/0
C    30.0.0.0/8 is directly connected, Serial3/0
S    40.0.0.0/8 [1/0] via 30.0.0.2
```

Figure4.2: Router1

```
Gateway of last resort is 30.0.0.1 to network 0.0.0.0
C    30.0.0.0/8 is directly connected, Serial2/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [1/0] via 30.0.0.1
```

Figure 4.3: Router2

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	

```
C:\>ping 40.0.0.3

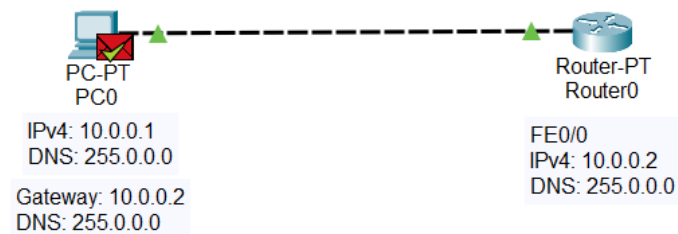
Pinging 40.0.0.3 with 32 bytes of data:

Reply from 40.0.0.3: bytes=32 time=35ms TTL=125
Reply from 40.0.0.3: bytes=32 time=37ms TTL=125
Reply from 40.0.0.3: bytes=32 time=24ms TTL=125
Reply from 40.0.0.3: bytes=32 time=38ms TTL=125

Ping statistics for 40.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 24ms, Maximum = 38ms, Average = 33ms
```

LABORATORY PROGRAM – 5

To understand the operation of TELNET by accessing the router in server room from a PC in IT office.



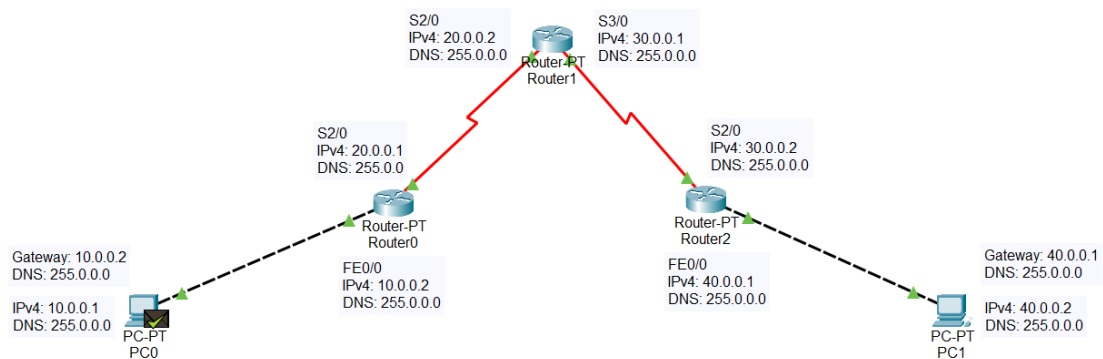
```
Router0
Physical Config CLI
IOS Command Line Interface
Router(config)#interface FastEthernet0/0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-6-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
ip address 10.0.0.2 255.0.0.0
Router(config-if)#exit
Router(config)#hostname R1
R1(config)#enable secret P0
R1(config)#line vty 0 5
R1(config-line)#login
% Login disabled on line 132, until 'password' is set
% Login disabled on line 133, until 'password' is set
% Login disabled on line 134, until 'password' is set
% Login disabled on line 135, until 'password' is set
% Login disabled on line 136, until 'password' is set
% Login disabled on line 137, until 'password' is set
R1(config-line)#password P1
R1(config-line)#exit
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console
R1#
R1#wr
Building configuration...
[OK]
R1#
```

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Router0	ICMP		0.000	N	0	(edit)	

```
PC0
Physical Config Desktop Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open
User Access Verification
Password:
R1>enable
Password:
R1#
```

LABORATORY PROGRAM – 6

Demonstrate the TTL/ Life of a Packet.



PDU Information at Device: Router0

OSI Model [Inbound PDU Details](#) Outbound PDU Details

PDU Formats

Ethernet II		Bytes	
PREAMBLE: 10101010	DEST ADDR: 000D.BD27.5B45		
SRC ADDR: 00D0.979D.0000	TYPE: 0x0800	DATA (VARIABLE LENGTH)	FCS: 0x00000000
IP			
VER: 4	IHL: 5	DSCP: 0x00	TL: 28
ID: 0x0004		FLAGS: 0x0	FRAG OFFSET: 0x000
TTL: 255	PRO: 0x01	CHKSUM	
SRC IP: 10.0.0.1			
DST IP: 40.0.0.2			
DATA (VARIABLE LENGTH)			
ICMP			

Figure 6.1: Inbound PDU, Router0

PDU Information at Device: Router0

OSI Model Inbound PDU Details [Outbound PDU Details](#)

PDU Formats

HDLC		Bits	
FLG: 0x7E	ADR: 0x8f	CONTROL: 0x0000	
DATA (VARIABLE LENGTH)			
FCS: 0x0000		FLG: 0x7E	
IP			
VER: 4	IHL: 5	DSCP: 0x00	TL: 28
ID: 0x0004		FLAGS: 0x0	FRAG OFFSET: 0x000
TTL: 254	PRO: 0x01	CHKSUM	
SRC IP: 10.0.0.1			
DST IP: 40.0.0.2			
DATA (VARIABLE LENGTH)			

Figure 6.2: Outbound PDU, Router0

PDU Information at Device: Router1

OSI Model [Inbound PDU Details](#) Outbound PDU Details

PDU Formats

HDLC		Bits	
FLG: 0x7E	ADR: 0x8f	CONTROL: 0x0000	
DATA (VARIABLE LENGTH)			
FCS: 0x0000		FLG: 0x7E	
IP			
VER: 4	IHL: 5	DSCP: 0x00	TL: 28
ID: 0x0004		FLAGS: 0x0	FRAG OFFSET: 0x000
TTL: 254	PRO: 0x01	CHKSUM	
SRC IP: 10.0.0.1			
DST IP: 40.0.0.2			
DATA (VARIABLE LENGTH)			

Figure 6.3: Inbound PDU, Router1

PDU Information at Device: Router1

OSI Model Inbound PDU Details [Outbound PDU Details](#)

PDU Formats

HDLC		Bits	
FLG: 0x7E	ADR: 0x8f	CONTROL: 0x0000	
DATA (VARIABLE LENGTH)			
FCS: 0x0000		FLG: 0x7E	
IP			
VER: 4	IHL: 5	DSCP: 0x00	TL: 28
ID: 0x0004		FLAGS: 0x0	FRAG OFFSET: 0x000
TTL: 253	PRO: 0x01	CHKSUM	
SRC IP: 10.0.0.1			
DST IP: 40.0.0.2			
DATA (VARIABLE LENGTH)			

Figure 6.4: Outbound PDU, Router1

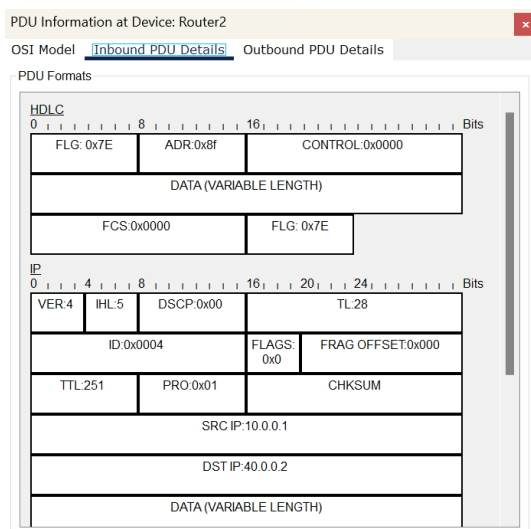


Figure 6.5: Inbound PDU, Router2

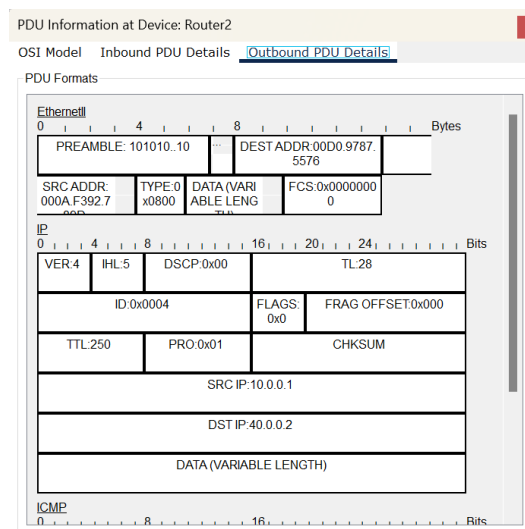


Figure 6.6: Outbound PDU, Router2

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC1	ICMP		0.000	N	0	(edit)	

```
C:\>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=72ms TTL=123
Reply from 40.0.0.2: bytes=32 time=53ms TTL=123
Reply from 40.0.0.2: bytes=32 time=55ms TTL=123
Reply from 40.0.0.2: bytes=32 time=69ms TTL=123

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 53ms, Maximum = 72ms, Average = 62ms
```


LABORATORY PROGRAM – 7(A)

To Configure IP addresses of the host using DHCP server within a LAN.

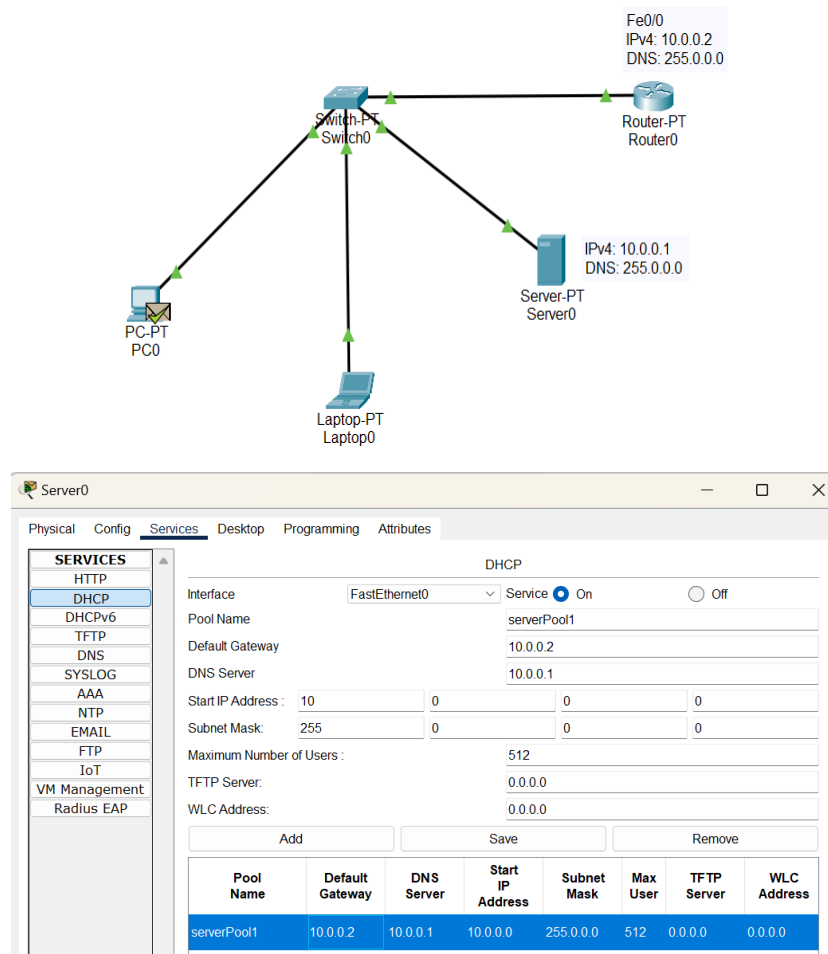


Figure 7.1: DHCP Service, Server0

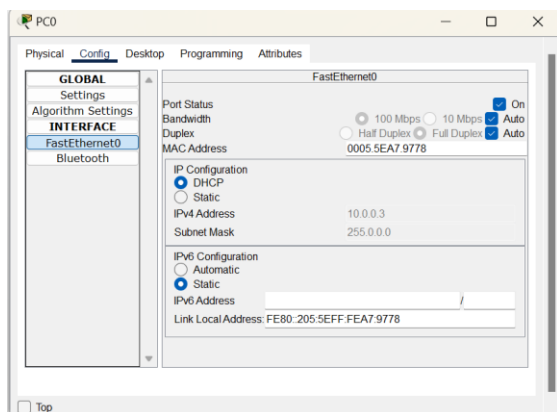


Figure 7.2: DHCP Service, PC0

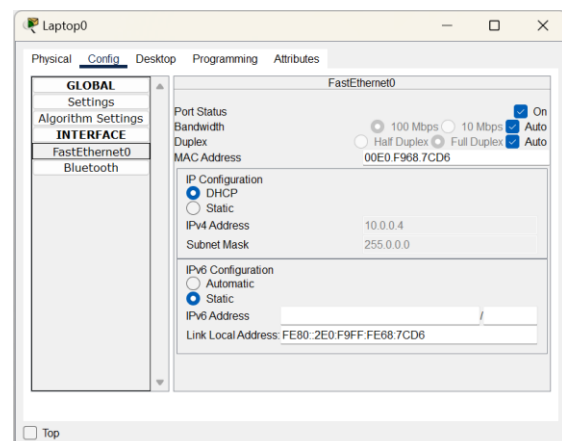





Figure 7.3: DHCP Service, Laptop0

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	

 PC0

Physical
Config
Desktop
Programming
Attributes

Command Prompt

```

Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

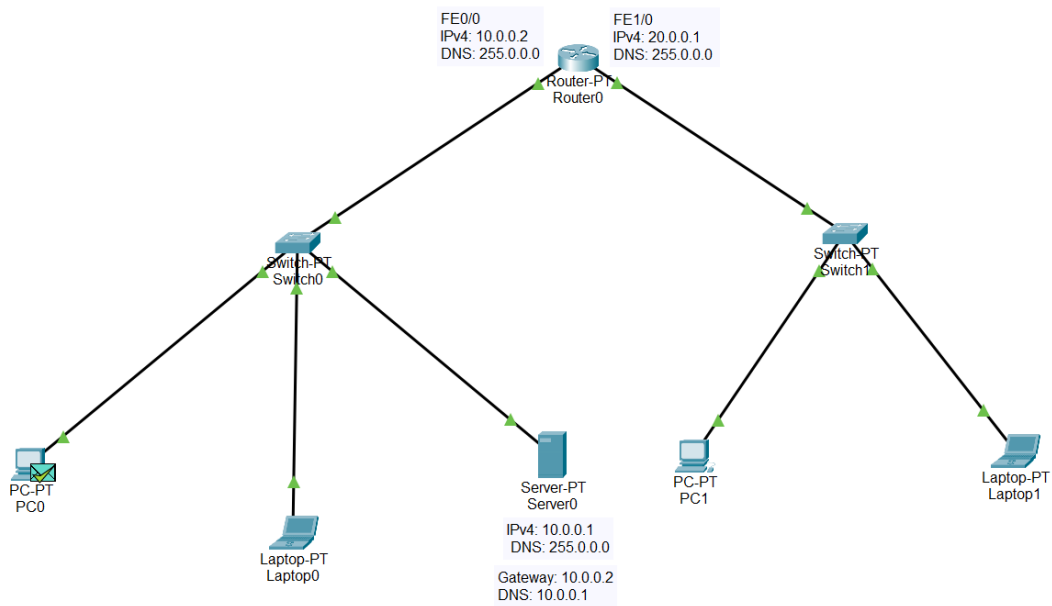
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

LABORATORY PROGRAM – 7(B)

To Configure IP addresses of the host using DHCP server outside a LAN.



Server0

Physical Config **Services** Desktop Programming Attributes

SERVICES

- HTTP
- DHCP**
- DHCPv6
- TFTP
- DNS
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP
- IoT
- VM Management
- Radius EAP

DHCP

Interface: FastEthernet0 Service: ☒ On ☐ Off

Pool Name: serverPool1

Default Gateway: 10.0.0.2

DNS Server: 10.0.0.1

Start IP Address: 10.0.0.0

Subnet Mask: 255.0.0.0

Maximum Number of Users: 512

TFTP Server: 0.0.0.0

WLC Address: 0.0.0.0

Add Save Remove

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool1	10.0.0.2	10.0.0.1	10.0.0.0	255.0.0.0	512	0.0.0.0	0.0.0.0
serverPool2	20.0.0.1	10.0.0.1	20.0.0.0	255.0.0.0	512	0.0.0.0	0.0.0.0
serverPool	0.0.0.0	0.0.0.0	10.0.0.0	255.0.0.0	512	0.0.0.0	0.0.0.0

Figure 7.2.1: DHCP Service, Server0

PC0

Physical Config **Desktop** Programming Attributes

GLOBAL

Settings

Algorithm Settings

INTERFACE

FastEthernet0

Bluetooth

Port Status: ☒ On

Bandwidth: ☒ Auto

Duplex: ☒ Auto

MAC Address: 0005.5EA7.9778

IP Configuration

☒ DHCP

☐ Static

IPv4 Address: 10.0.0.3

Subnet Mask: 255.0.0.0

IPv6 Configuration

☐ Automatic

☒ Static

IPv6 Address:

Link Local Address: FE80:205:5EFF:FEA7:9778

Figure 7.2.2: DHCP Service, PC0

Laptop0

Physical Config **Desktop** Programming Attributes

GLOBAL

Settings

Algorithm Settings

INTERFACE

FastEthernet0

Bluetooth

Port Status: ☒ On

Bandwidth: ☒ Auto

Duplex: ☒ Auto

MAC Address: 00E0.F968.7CD6

IP Configuration

☒ DHCP

☐ Static

IPv4 Address: 10.0.0.4

Subnet Mask: 255.0.0.0

IPv6 Configuration

☐ Automatic

☒ Static

IPv6 Address:

Link Local Address: FE80:2E0:F9FF:FE68:7CD6

Figure 7.2.3: DHCP Service, Laptop0

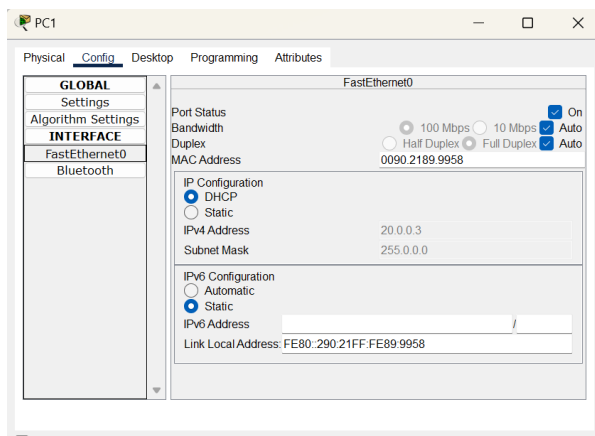


Figure 7.2.4: DHCP Service, PC1

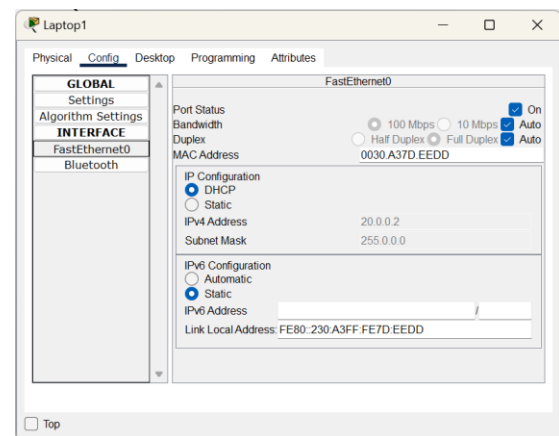


Figure 7.2.5: DHCP Service, Laptop1

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	
	Successful	PC1	Laptop1	ICMP		0.004	N	1	(edit)	

LABORATORY PROGRAM – 8

To Configure DNS server to demonstrate the mapping of IP addresses and Domain names.

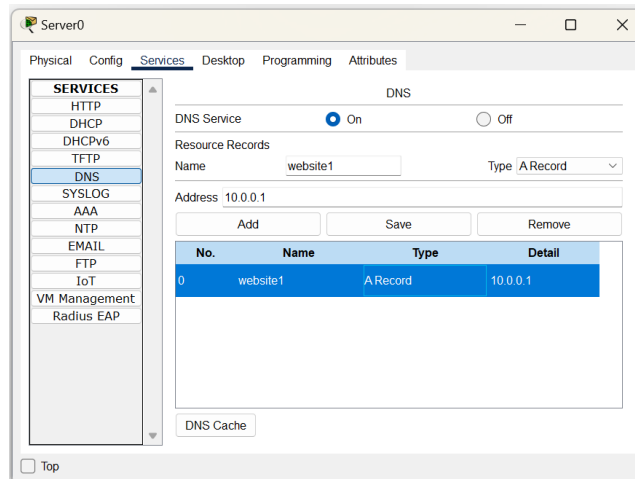
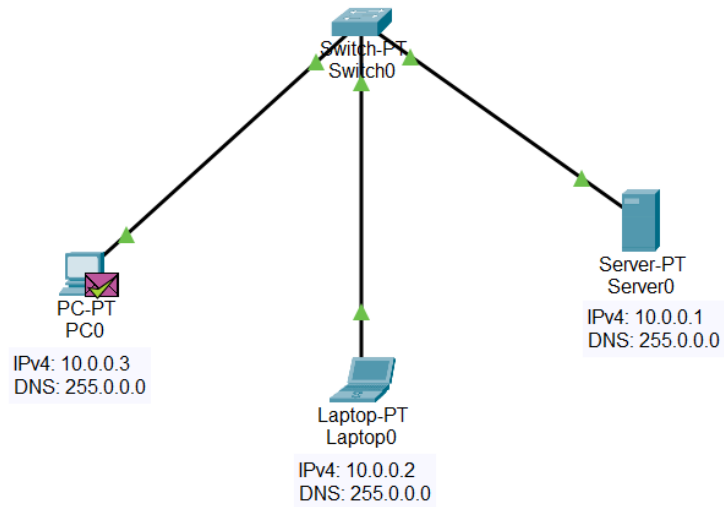


Figure 8.1: DNS Service, Server0

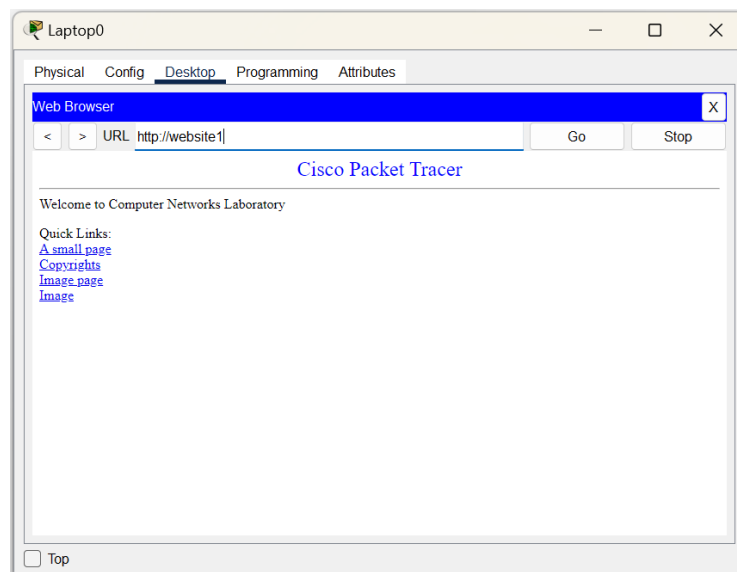


Figure 8.2: DNS Service, Laptop0

LABORATORY PROGRAM – 9

To Configure RIP routing protocol in Routers.

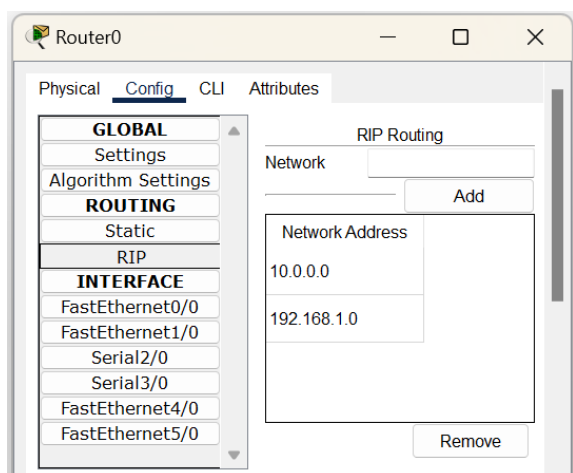
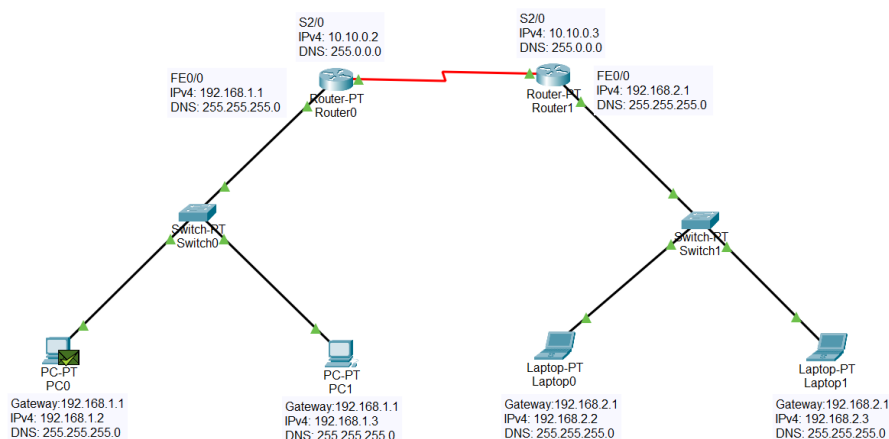


Figure 9.1: RIP, Router0

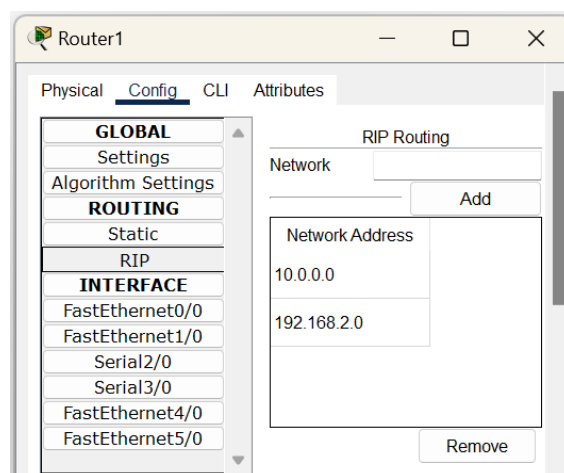


Figure 9.2: RIP, Router

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop1	ICMP		0.000	N	0	(edit)	

```
C:\>ping 192.168.2.3
```

```
Pinging 192.168.2.3 with 32 bytes of data:
```

```
Reply from 192.168.2.3: bytes=32 time=18ms TTL=126
Reply from 192.168.2.3: bytes=32 time=14ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126
```

```
Ping statistics for 192.168.2.3:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 18ms, Average = 8ms
```

LABORATORY PROGRAM – 10

To demonstrate communication between two devices using a wireless LAN.

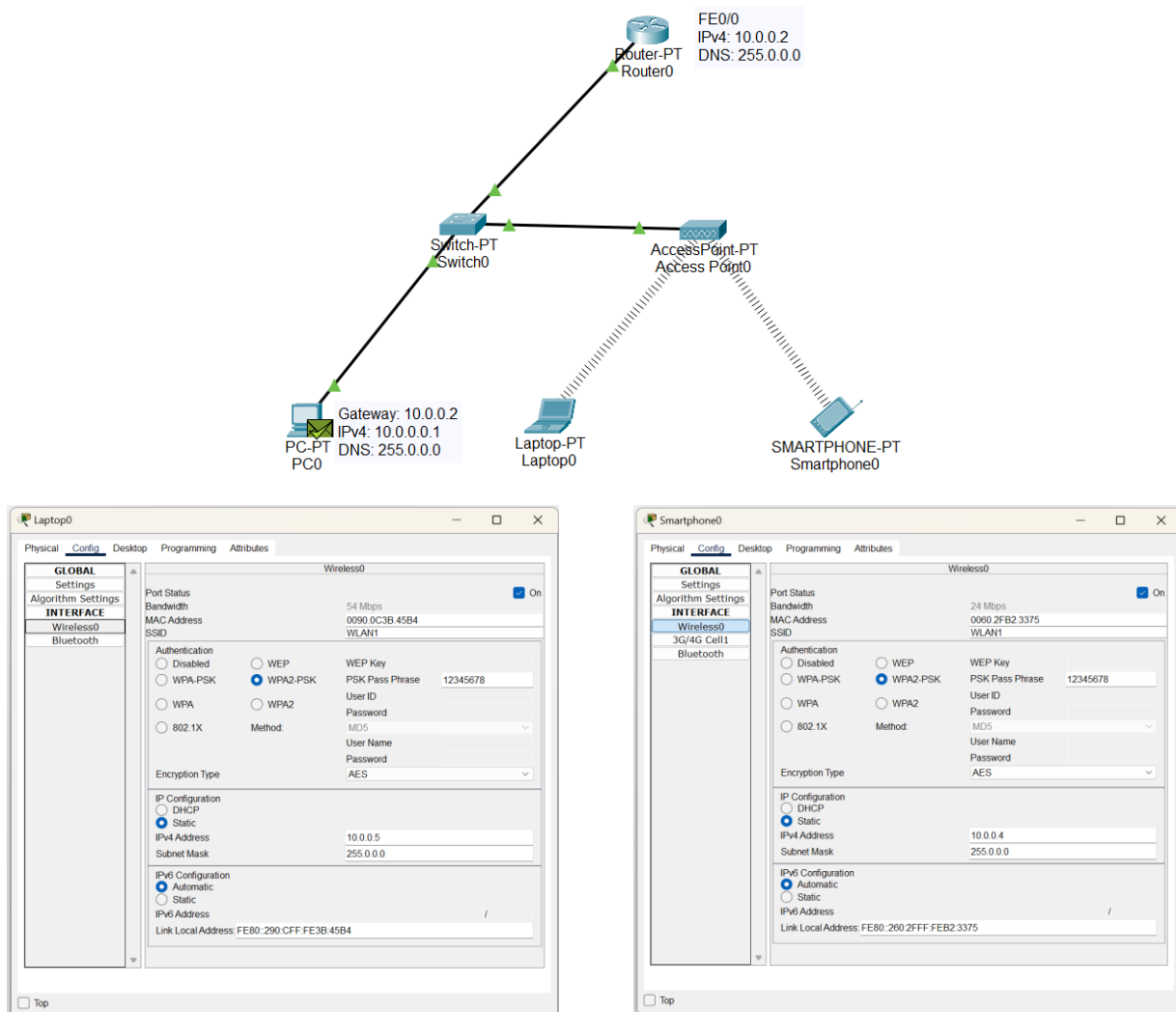


Figure 10.1: Laptop0, Wireless0

Figure 10.2: Smartphone0, Wireless0

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	

```

PC0
Physical Config Desktop Programming Attributes
Command Prompt

Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data:

Reply from 10.0.0.5: bytes=32 time=8ms TTL=128
Reply from 10.0.0.5: bytes=32 time=28ms TTL=128
Reply from 10.0.0.5: bytes=32 time=30ms TTL=128
Reply from 10.0.0.5: bytes=32 time=36ms TTL=128

Ping statistics for 10.0.0.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 8ms, Maximum = 36ms, Average = 25ms
    
```

LABORATORY PROGRAM – 11

To demonstrate the working of Address Resolution Protocol (ARP) within a LAN for communication.

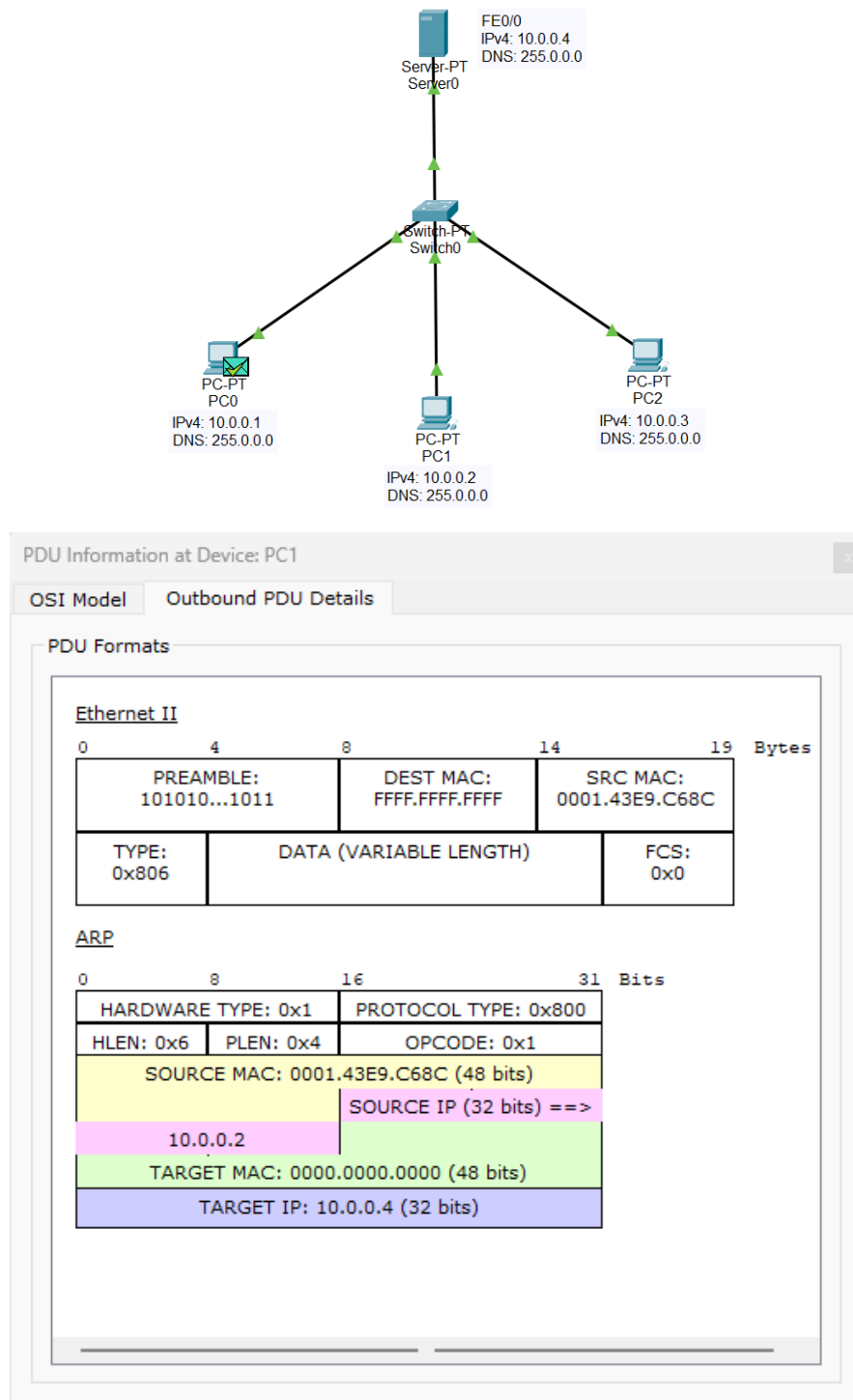


Figure 11.1: Inbound ARP, PC1

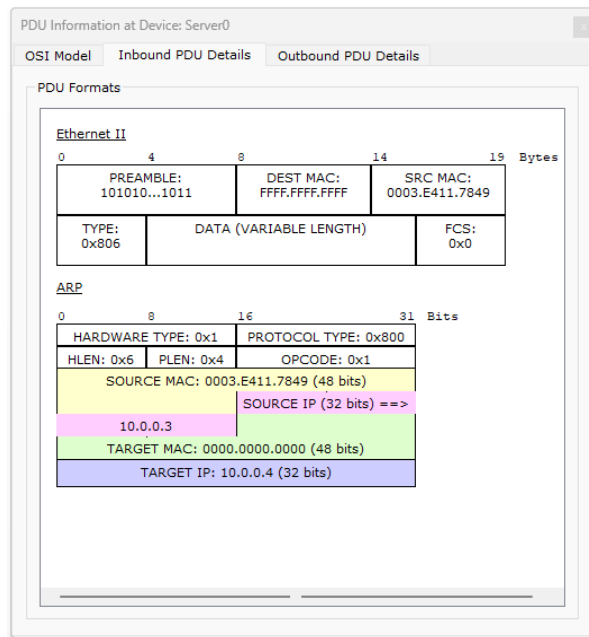


Figure 11.2: Inbound ARP, Server0

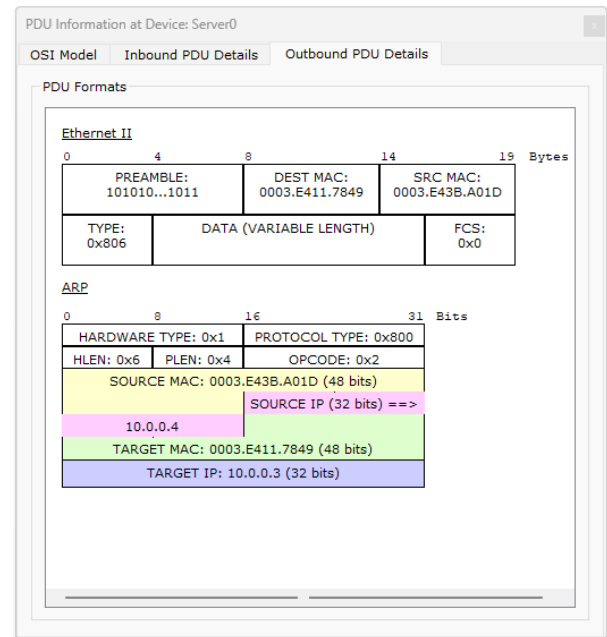


Figure 11.3: Outbound ARP, Server0

ARP Table for Server0

IP Address	Hardware Address	Interface
10.0.0.1	00E0.B062.0C32	FastEthernet0
10.0.0.2	0001.43E9.C68C	FastEthernet0

Figure 11.4: ARP Table, Server0

ARP Table for PC1

IP Address	Hardware Address	Interface
10.0.0.4	0003.E43B.A01D	FastEthernet0

Figure 11.5: ARP Table, PC1

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Server0	ICMP		0.000	N	0	(edit)	

PC0

Physical Config Desktop Programming Attributes

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=23ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 23ms, Average = 5ms
```

LABORATORY PROGRAM – 12

To create a VLAN on top of the physical LAN and enable communication between physical LAN and virtual LAN.

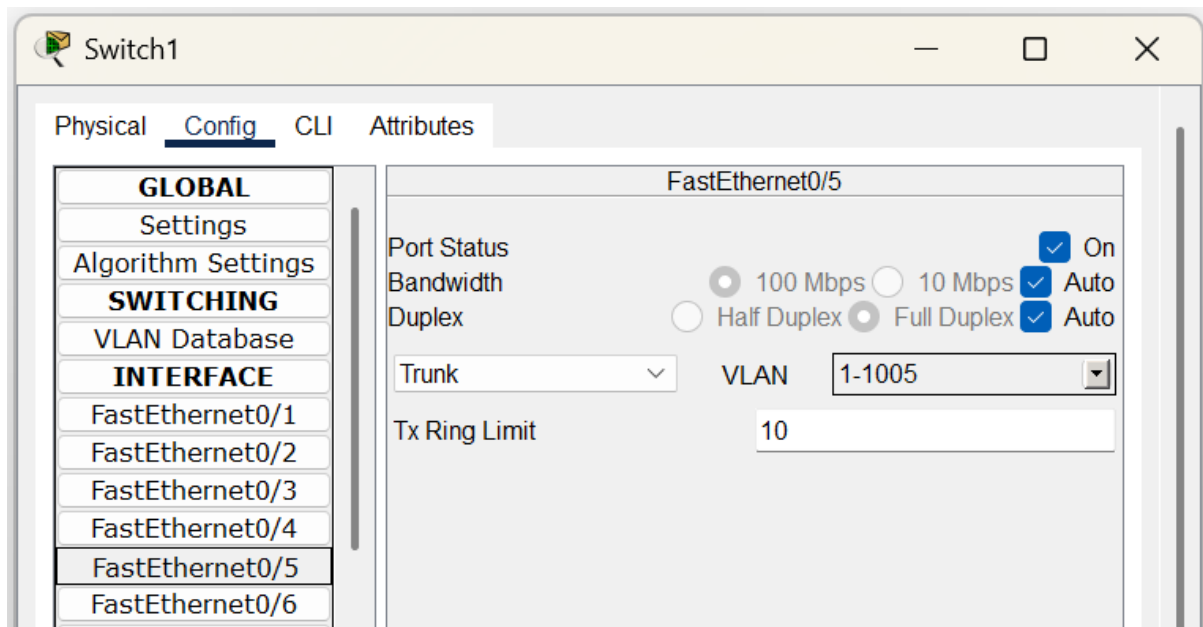
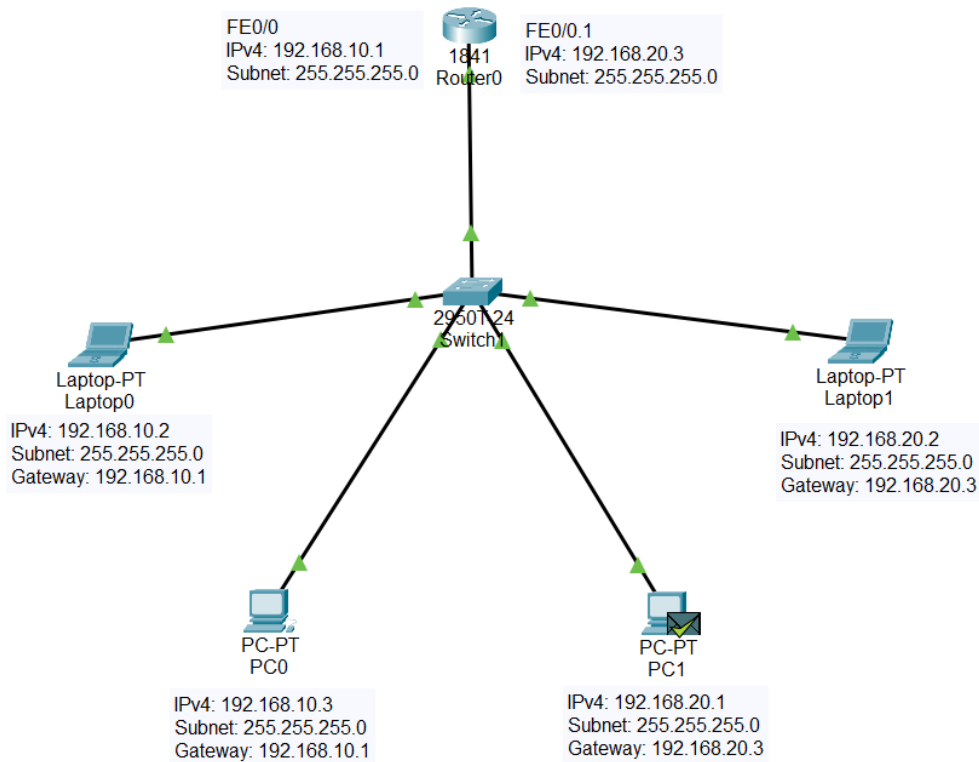


Figure 12.1: FE0/5 Switchport Trunk

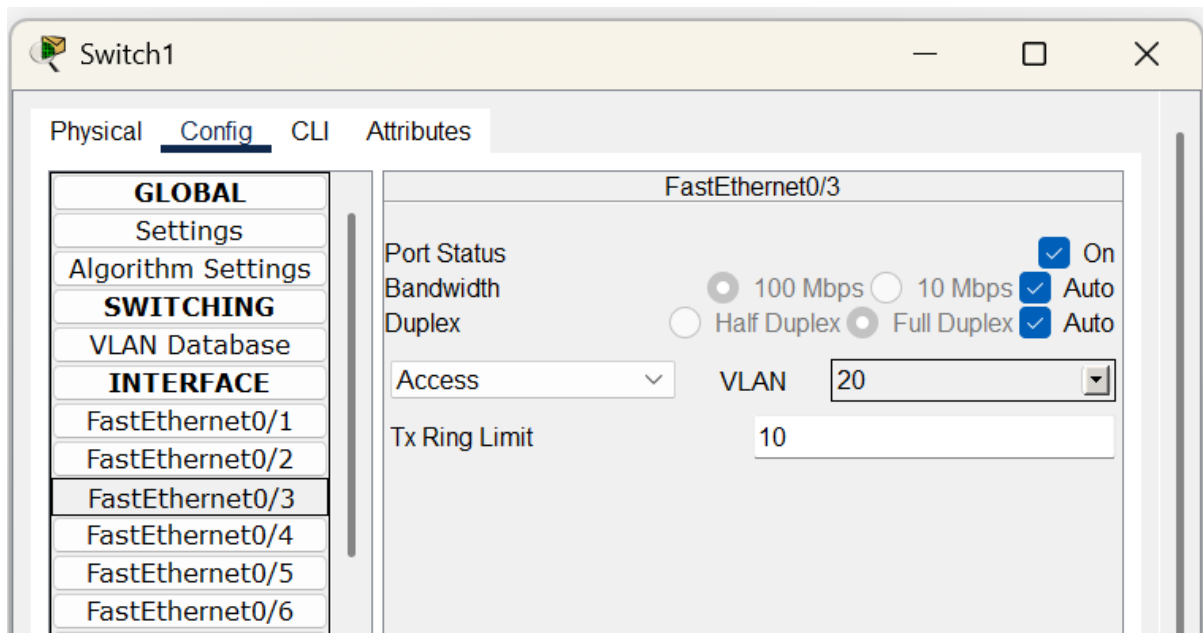


Figure 12.2: FE0/3 Switchport Access

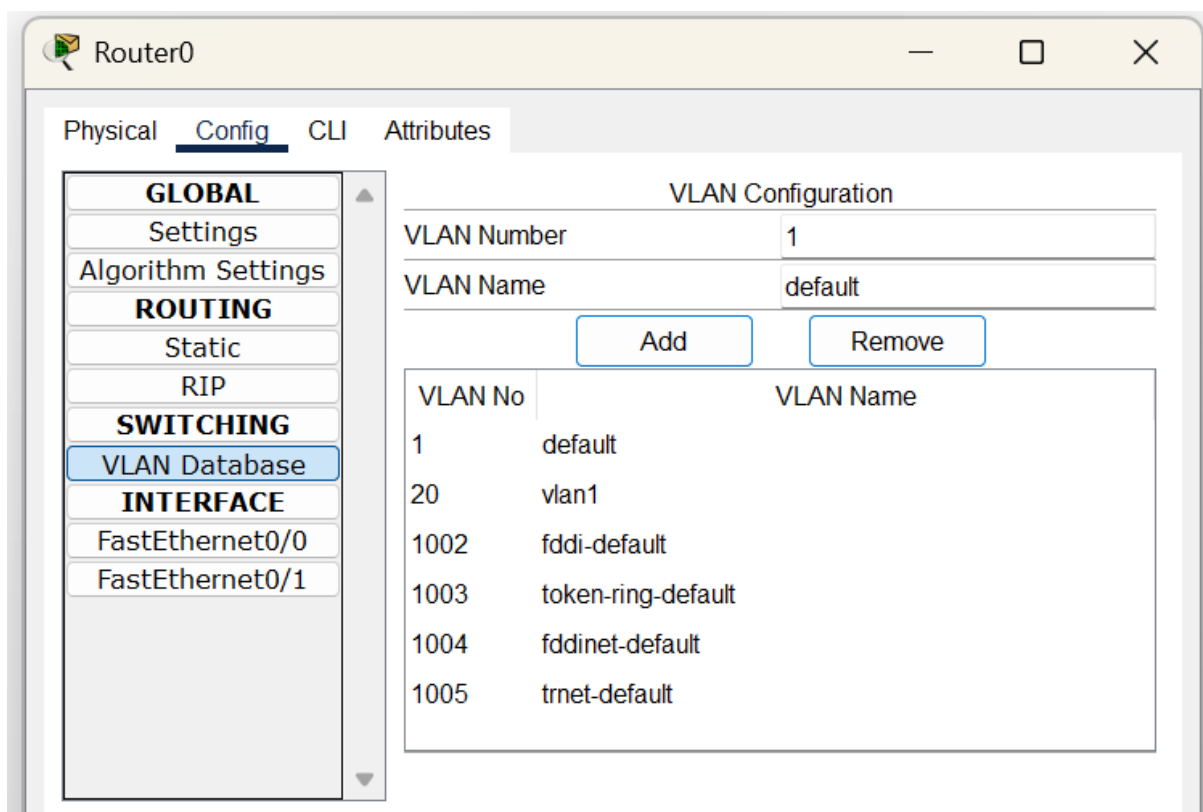


Figure 12.3: Router0 VLAN Database

```
Router(config)#interface FastEthernet0/0.1
Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip address 192.168.20.3 255.255.255.0
Router(config-subif)#no shutdown
```

Figure 3: Router0, FE0/0.1

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC1	Router0	ICMP		0.000	N	0	(edit)	

```
C:\>ping 192.168.20.3
```

```
Pinging 192.168.20.3 with 32 bytes of data:
```

```
Reply from 192.168.20.3: bytes=32 time=2ms TTL=255
```

```
Reply from 192.168.20.3: bytes=32 time<1ms TTL=255
```

```
Reply from 192.168.20.3: bytes=32 time<1ms TTL=255
```

```
Reply from 192.168.20.3: bytes=32 time<1ms TTL=255
```

```
Ping statistics for 192.168.20.3:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

LABORATORY PROGRAM – 13

Write a program for error detecting code using CRC-CCITT (8-bits).

Code

```
def xor(dividend, divisor):
    """Perform XOR operation between
    dividend and divisor."""
    result = ""
    for i in range(1, len(divisor)):
        result += '0' if dividend[i] ==
        divisor[i] else '1'
    return result

def crc(data, gen_poly):
    """Compute the CRC check value using
    CRC-CCITT (8-bit)."""
    data_length = len(data)
    gen_length = len(gen_poly)

    # Append n-1 zeros to the data
    padded_data = data + '0' * (gen_length -
    1)

    check_value =
    padded_data[:gen_length]

    for i in range(data_length):
        if check_value[0] == '1':
            # XOR operation if the first bit is 1
            check_value = xor(check_value,
            gen_poly)
        else:
            # Retain original check value if
            first bit is 0
            check_value = check_value[1:]

    # Shift left and add the next data bit
    if i + gen_length < len(padded_data):
        check_value += padded_data[i +
        gen_length]

    return check_value[1:] # Remove the
    leading bit

def receiver(data, gen_poly):
    """Simulate the receiver side to check
    for errors."""
    print("\n-----")
    print("Data received:", data)

    # Perform CRC computation on
    received data
    remainder = crc(data, gen_poly)

    # Check if the remainder is all zeros
    if '1' in remainder:
        print("Error detected")
    else:
        print("No error detected")

if __name__ == "__main__":
    # Input data and generator polynomial
    data = input("Enter data to be
    transmitted: ")
    gen_poly = input("Enter the Generating
    polynomial: ")
```

```

# Compute CRC check value
check_value = crc(data, gen_poly)
print("\n-----
--")

print("Data padded with n-1 zeros:",
data + '0' * (len(gen_poly) - 1))

print("CRC or Check value is:",
check_value)

# Append check value to data for
transmission

transmitted_data = data + check_value
print("Final data to be sent:",
transmitted_data)

print("-----
\n")

# Simulate the receiver side
received_data = input("Enter the
received data: ")

receiver(received_data, gen_poly)

```

Output

```

Enter data to be transmitted: 1001100
Enter the Generating polynomial: 100001011

-----
Data padded with n-1 zeros: 100110000000000
CRC or Check value is: 0100010
Final data to be sent: 10011000100010
-----

Enter the received data: 10011000100011

-----
Data received: 10011000100011
Error detected

```

LABORATORY PROGRAM – 14

Write a program for congestion control using Leaky bucket algorithm.

Code

```
# Getting user inputs
storage = int(input("Enter initial packets in the bucket: "))
no_of_queries = int(input("Enter total no. of times bucket content is checked: "))
bucket_size = int(input("Enter total no. of packets that can be accommodated in the bucket: "))
input_pkt_size = int(input("Enter no. of packets that enters the bucket at a time: "))
output_pkt_size = int(input("Enter no. of packets that exits the bucket at a time: "))

for i in range(no_of_queries): # space left
    size_left = bucket_size - storage
    if input_pkt_size <= size_left:
        # update storage
        storage += input_pkt_size
    else:
        print("Packet loss =", input_pkt_size)

print(f"Buffer size = {storage} out of bucket size = {bucket_size}")

# as packets are sent out into the network, the size of the storage decreases
storage -= output_pkt_size
```

Output

```
Enter initial packets in the bucket: 0
Enter total no. of times bucket content is checked: 4
Enter total no. of packets that can be accommodated in the bucket: 10
Enter no. of packets that enters the bucket at a time: 4
Enter no. of packets that exits the bucket at a time: 1
Buffer size = 4 out of bucket size = 10
Buffer size = 7 out of bucket size = 10
Buffer size = 10 out of bucket size = 10
Packet loss = 4
Buffer size = 9 out of bucket size = 10
```